EXHIBIT 10

DOCKET NO: 0100157-00245

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT: 7,945,539

INVENTOR: DAVID A. FARBER
AND RONALD D. LACHMAN

FILED: OCT. 31, 2007 ISSUED: MAY 17, 2011

TITLE: DISTRIBUTING AND
ACCESSING DATA IN A DATA
PROCESSING SYSTEM

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR *INTER PARTES REVIEW* OF U.S. PATENT NO. 7,945,539 UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104

TABLE OF CONTENTS

I. MANDATORY NOTICES

A. Real Parties-in-Interest

EMC Corporation ("Petitioner") is the real parties-in-interest.

B. Related Matters

The '539 patent is one of an extensive patent family of continuation and divisional applications. Ex. 1008 shows the patent family, with patents in red and blue including the '539 patent being asserted in the litigation *PersonalWeb Technologies LLC v. EMC Corporation and VMware, Inc.* (No. 6:11-cv-00660-LED) (E.D. Tex.), filed on December 8, 2011 and served on December 16, 2011.

Petitioner is also seeking *Inter Partes* Review of related U.S. Patents Nos. 5,978,791, 6,415,280, 7,945, 544, 7,949,662, and 8,001,096, and requests that they be assigned to the same Board for administrative efficiency.

C. Counsel

Lead Counsel: Peter M. Dichiara (Registration No. 38,005)

Backup Counsel: David L. Cavanaugh (Registration No. 36,476)

D. Service Information

Email: Peter Dichiara, peter.dichiara@wilmerhale.com

Post and Hand Delivery: WilmerHale, 60 State St., Boston MA 02109

Telephone: 617-526-6466 Facsimile: 617-526-5000

E. Certification of Grounds for Standing

Petitioner certifies pursuant to Rule 42.104(a) that the patent for which review is sought is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

II. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

A. Prior Art Patents and Printed Publications

Pursuant to Rules 42.22(a)(1) and 42.104 (b)(1)-(2), Petitioner challenges claims 10, 21, and 34 of U.S. Patent No. 7,945,539 ("the '539 patent", Ex. 1001) as anticipated by or unpatentable in view of the following patents and printed publications:

1. S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," University of Tennessee Technical Report CS-95-278 (Feb. 1995) ("Browne ", Ex. 1002).[1]

[1]The Browne February 1995 publication qualifies as prior art under 35 U.S.C. § 102(a), and is used in this petition because it includes illustrations which facilitate explanation of the grounds of the invalidity. Petitioner also has attached as exhibits and included in its claim charts two earlier versions of this publication – S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," http://www.netlib.org/utk/papers/lifn/main.html (Nov. 11, 1994) (Ex. 1006); and K. Moore et al., "An Architecture for Bulk File Distribution,"

2. Albert Langer, "Re: dl/describe (File descriptions)," article <1991Aug7.225159.786@newshost.anu.edu.au> in Usenet newsgroups "alt.sources.d" and "comp.archives.admin" (August 7, 1991) ("Langer", Ex. 1003)[2]
3. Kantor, "The Frederick W. Kantor Contents-Signature System Version 1.22," FWKCS122.REF (August 10, 1993) ("Kantor", Ex.

Network Working Group Internet Draft (July 27, 1994) (Exhibit 1007). As Dr. Clark confirms in his declaration, the relevant disclosures are substantially the same. If the Patent Owner attempts to claim an earlier priority date of the challenged claims, Petitioner may rely on the earlier publications for invalidity , alone or in combination with the other references cited in this petition , alone or in combination with the other references cited in this petition.

[2] Langer was made available on the "alt.sources.d" and "comp.archives.admin" newsgroup distribution lists on August 7, 1991. Both newsgroups were widely disseminated and readily accessible to the relevant technical community. Specifically, the "alt.sources.d" newsgroup was devoted to technical discussions relating to the "alt.sources" source code repository. The "comp.archives.admin" newsgroup hosted discussions relating to computer archive administration. Therefore, an interested person would have been able to readily locate Langer among postings related to those subjects, both of which are in the same technical field as the '539 patent.

1004).[3]

4. Woodhill et al., U.S. Patent No. 5,649,196, entitled "System and Method For Distributed Storage Management on Networked Computer Systems Using Binary Object Identifiers," filed Nov. 9, 1995 as a continuation of application 85,596, filed July 1, 1993 ("Woodhill", Ex. 1005).
5. Fischer, U.S. Patent No. 5,475,826, entitled "Method for Protecting a Volatile File Using a Single Hash," filed Nov. 19, 1993 ("Fischer", Ex. 1036).

[3] Kantor's FWKCS user manual has been publicly and freely available continuously since August 1993. Kantor distributed the user manual with the FWKCS program as shareware and posted it online to electronic Bulletin Board Systems including "The Invention Factory" and "Channel 1" for an extended period of time, where it could be downloaded by anyone. As such the document was accessible to others in the relevant community of BBS users and system operators. (*See* Kantor at 3; *see also* 158-59; Ex. 1004.)

B. There is a Reasonable Likelihood that at least One Claim of the '539 Patent is Unpatentable Under 35 U.S.C. §§ 102, 103

Section VI below explains how the above-cited patents and printed publications create a reasonable likelihood that Petitioner will prevail with at least one of the challenged claims. *See* 35 U.S.C. § 314(a). Indeed, that section together with the attached claim charts of Exs. 1045-1048 and the Declaration of Dr. Douglas Clark, a Professor of Computer Science at Princeton University ("Clark Decl."; Ex. 1009), demonstrate that all of the challenged claims are anticipated by, or unpatentable in view of, each of these references.

C. Relief Requested

Petitioner requests cancellation of claims 10, 21, and 34, the challenged claims, as unpatentable under 35 U.S.C. §§ 102 and 103.

III. Claim Construction

The claims should be given their "broadest reasonable construction in light of the specification." 37 C.F.R. § 42.100(b). The claim terms can be understood by their plain and ordinary meanings except where construed in the specification. The specification includes the following relevant constructions:

Claim Term	Construction
"data" and "data item"	"as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned

Claim Term	Construction
	image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits." ('539 patent, col. 2, ll. 16-21, s*ee also* col. 2, l. 26-38 (indicating "data items" can include "files, directories, records in the database, objects in object-oriented programming, locations in memory or on a physical device, or the like"); '539 claim 9 (indicating "sequence[s] of bits" can include, in addition to the items articulated above, "a software product [or] a portion of a software product".); Ex. 1001.)
"file system"	"a collection of directories. A directory is a collection of named files – both data files and other directory files." ('539 patent, col. 5, ll. 61-63; Ex. 1001.)
"file"	"a named data item which is either a data file (which may be simple or compound) or a directory file. A simple file consists of a single data segment. A compound file consists of a sequence of data segments. A data segment is a fixed sequence of bytes." ('539 patent, col. 5, ll. 64- col. 6, ll. 1; Ex. 1001.)
"location"	"with respect to a data processing system, refers to any of a particular processor in the system, a memory of a particular processor, a storage device, a removable storage medium (such as a floppy disk or compact disk), or any other physical location in the system" ('539 patent, col. 6, ll. 12-17; Ex. 1001.)
"True Name, data identity, and data identifier"	"refer to the substantially unique data identifier for a particular data item" ('539 patent, col. 6, ll. 20-22, see also col. 15, l. 37 - col. 16, l. 8 (describing mechanism for calculating True Name using MD hash function); Ex. 1001.)

IV. OVERVIEW OF THE '539 PATENT

A. Brief Description

The '539 patent is directed to data storage systems that use "substantially unique data identifiers" – based on all the data in a data item and only the data in the data item – to identify and access data items. (*See, e.g.*, '539 patent, Title, Abstract, and col. 1, ll. 44-48; Ex. 1001.) The patent uses these identifiers to perform basic file management functions, such as requesting and obtaining files or other data items, and eliminating unwanted duplicate copies of data items—admittedly old problems. (*See, e.g.*, *id*. at col. 3, ll. 4-15, 30-39.)

According to the patent, prior art systems identified data items based on their location or address within the data processing system. (*Id*. at col. 1, ll. 56-58.) For example, files were often identified by their context or "pathname," that is, information specifying a path through the computer directories to the particular file (*e.g.*, C:\My Documents\Law School\1L\TortsOutline.txt). (*Id*. at col. 1, ll. 65 – col.2, ll. 5.) The patent contends that all prior art systems operated in this manner: "In ***all*** of the prior data processing systems the names or identifiers provided to identify data items. . . are ***always*** defined relative to a specific context," and "there is ***no*** direct relationship between the data names and the data item." (*Id*. at col. 2, ll. 26-31, ll. 39-40 (emphasis added).)

According to the patent, this prior art practice of identifying a data item by its context or pathname resulted in certain shortcomings. For example, with pathname identification, the same data name may refer to different data items, or conversely, two different data names may refer to the same data item. (*Id*. at col. 2, ll. 39-43.) Moreover, because there is no correlation between the contents of a data item and its pathname, there is no *a priori* way to confirm that the data item is in fact the one named by the pathname. (*Id*. at col. 2, ll. 44-47.) Furthermore, context or pathname identification may more easily result in the creation of unwanted duplicate data items, e.g., multiple copies of a file on a file server.[4] (*Id*. at col. 3, ll. 4-15.)

The '539 patent purports to address these shortcomings. (*Id*. at col. 3, ll. 30-48.) It suggests that "it is therefore desirable to have a mechanism . . . to determine a common and substantially unique identifier for a data item, using only the data in the data item and not relying on any sort of context." (*Id*. at col. 3, ll. 31-35.) Moreover, "[i]t is further desirable to have a mechanism for reducing multiple copies of data items . . . and to have a mechanism which enables the identification of identical data items so as to reduce multiple copies." (*Id*. at col. 3, ll. 36-39.)

---

[4] For example, Alice and Bob both download the same copy of the James Bond movie *Goldfinger*. Alice saves her copy at "C:\Movies\Bond\Goldfinger.mov", and Bob saves his copy at "C:\Videos\007\Bond-Goldfinger.mov".

To do so, the '539 patent provides data identifiers that "depend[] on all of the data in the data item and only on the data in the data item." ('539 patent, col. 1, ll. 44-48, col. 3, ll. 54-55; Ex. 1001.) The preferred embodiments use the well-known MD5 or SHA message digest hash functions[5] to calculate a substantially unique identifier from the contents of the data item. (*Id*. at col. 14, l. 24 – col. 16, l. 8.) The system first computes the 16-byte (128-bit) message digest of the data item and then appends the size of the data item to produce a 160-bit identifier. (*Id*. at Fig. 10A and col. 15, ll. 42-48.) The patent calls these context- or location-independent, content-based identifiers a "True Name" – a phrase admittedly "coined by the inventors." (U.S. Patent No. 6,415,280 Prosecution History, Response (Aug. 22, 2001), at 22; Ex. 1019.)

If a data item is large, it may include multiple components or "segments," representing the larger "compound data item." ('539 patent, col. 15, ll. 49-59; Ex. 1001.) A True Name identifier can be computed for each of the segments based on a hash of the contents of the segment. (*Id.* at col. 15, ll. 53-62.) Together,

---

[5] A message digest or hash function transforms of a piece of data into a much shorter form by performing mathematical operations on its content. (*See, e.g.*, D. Banisar et al., The Third CSPR Cryptography and Privacy Conference at 509 (1993) ; Ex. 1010.) The '539 patent admits that message digest functions were known. ('539 patent, col. 14, ll. 46-50; Ex. 1001.)

these segment identifiers form an "indirect block." (*Id*. at col. 15, ll. 59-62.) A True Name identifier is then computed for the compound data item as a whole based on a hash of the contents of the indirect block (*i.e*., a hash of the segment hashes). (*Id*. at col. 15, ll. 62-67.)

With these identifiers, the patent asserts, "data items can be accessed by reference to their identities (True Names) independent of their present location." (*Id*. at col. 35, ll. 39-41; *see also* col. 35, ll. 60-62.) The actual data item corresponding to these location-independent identifiers may reside anywhere, e.g., locally, remotely, offline. (*Id*. at col. 35, ll. 41-43.) "Thus, the identity of a data item is independent of its name, origin, location, address, or other information not derivable directly from the data, and depends only on the data itself." (*Id*. at col. 3, ll. 55-58.)

In the preferred embodiments, the substantially unique identifiers are used to "augment" standard file management functions of an existing operating system. ('539 patent at col. 6, ll. 25-32; Ex. 1001.) For example, a local directory extensions (LDE) table[6] is indexed by a pathname or contextual name of a file and

[6] According to the patent, an LDE table is a data structure which provides information about files and directories in the system and includes information in addition to that provided by the native file system. ('539 patent, col. 8, ll. 28-36;

also includes True Names for most files. (*Id*. at col. 8, ll. 28-36.) A True File registry (TFR) lists True Names, and stores "location, dependency, and migration information about True Files." (*Id*. at col. 8, ll. 37-39, 42-44.) True Files are identified in the True File registry by their True Names, and can be looked up in the registry by their True Names. (*Id*. at col. 8, ll. 40-42; col. 25, ll. 33-34.) This look-up provides, for each True Name, a list of the locations, such as file servers, where the corresponding file is stored. (*Id*. at col. 35, ll. 47-49; *see also* col. 25, ll. 49-54.)

When opening or reading a file, the "Read File" mechanism "is aware of compound files and indirect blocks, and it uses [other] mechanisms to make sure that component segments are locally available. . . . When [a compound file] is opened only its indirect block is copied. When the corresponding file is read, the required component segments are realized and therefore copied." (*Id*. at col. 35, ll. 28-38.)

B. The Prosecution History of the '539 Patent

The '539 patent is based on an application that was originally filed on April 11, 1995. Initial claim 10 of the '539 patent, for example, reads as follows:

Ex. 1001.)

> 10. A method of obtaining a data item at a computer in a network of computers, said data item comprising a plurality of segments, the method comprising:
>
> (A) for each particular segment of said plurality of segments that comprise said data item, said computer:
>
> (a1) requesting said particular segment from at least one of a plurality of computers in said network of computers; and
>
> (a2) obtaining said particular segment from said at least one of a plurality of computers in said network of computers.

(Application as Filed on Oct. 31, 2007, at 74; Ex. 1024**.)** This claim required little more than requesting and obtaining segmented data items.

All claims were rejected as anticipated by U.S. Pat. No. 5,649,196 to Woodhill et al. (Office Action, May 6, 2009, at 2; Ex. 1025**)**. For example, with regard to then-pending claim 10, the office action stated:

> **As per claim 10**, Woodhill teaches a method of obtaining a data item at a computer in a network of computers, said data item comprising a plurality of segments (Col. 4 lines 20-35 and Fig. 5A, element 136), the method comprising:
>
> - "for each particular segment of said plurality of segments that comprise said data item, "said computer: (a1) requesting said particular segment from at least one of a plurality of computers in said network of computers" at Col. 13 lines 1-20 and Col. 14 lines 1-25;
> - "(a2) obtaining said particular segment from said at least one of a plurality of computers in said network of computers" at Col. 13 lines 1-20 and Col. 14 lines 1-25.

(*Id*. at 5.) In response, the applicants amended the claims to recite that the segments are stored among computers of a peer-to-peer network, and that different segments are obtained from different computers. (Amend., Oct. 5, 2009, at 4-5; Ex. 1026.) In conjunction with these amendments, the applicants argued that Woodhill lacked any teaching or suggestion of a peer-to-peer networks (*see, e.g.*, *id*. at 17 ("there is nothing in Woodhill to teach or in any way suggest that any of the computers across any local area network form a peer-to-peer network.")), and that Woodhill lacked any teaching of obtaining different segments from different computers. (*Id*. at 17-18.)

The examiner specifically rejected applicants' argument concerning peer-to-peer networks (Office Action, Jan. 12, 2010, at 17; Ex. 1027), and again rejected all of the claims. (*Id*. at 2 and 7.) In addressing amended claim 10, the office action relied on the new combination of Gardner et al. (U.S. Pat. No. 5,649,196) in view of Kindell et al. (U.S. Pat. No. 5,630,067). (Office Action, Jan. 12, 2010, at 7; Ex. 1027.) The applicants attempted to traverse this new rejection but failed (Amend. after Final, Mar. 12, 2010; Ex. 1028; Advisory Action, Mar. 23, 2010; Ex. 1029), and pursued a request for continued examination which made substantial amendments to the claim language yet again. (Request for Continued Examination, Oct. 14, 2010; Ex. 1030.) For example, claim 10 was amended to remove the limitations that were added in the prior amendment but failed, and the

claims also were amended to include completely different limitations, namely, "segment identifiers" based on a given function of the data and only the data of the particular segment, and a "first identifier" based on a function of the segment identifiers:

> 10. (Currently amended) A computer-implemented method of obtaining access to a data item at a first computer in a ~~peer-to-peer (P2P)~~ network of computers, said data item comprising a plurality of segments, ~~each of said plurality of segments being stored on more than one computer in said P2P network,~~ the method comprising the steps of:
>
> (A) in response to a request, said request comprising a first identifier, obtaining a plurality of segment identifiers, each of said segment identifiers corresponding to one of said plurality of segments, the segment identifier for each particular segment being based, at least in part, on a first given function of the data

comprising said particular segment and only the data in said particular segment, where any two identical segments will have identical segment identifiers as determined using said first given function, wherein said first identifier is based, at least in part, on a second given function of the plurality of segment identifiers; ~~obtaining information about computers in said P2P network that may have a copy of each particular segment of said plurality of segments that comprise said data item;~~

(B) by hardware in combination with software, using ~~based~~ at least ~~in part on~~ one of said segment identifiers obtained in step (A) ~~information~~, requesting at least one particular segment of said plurality of segments that comprise said data item from ~~a plurality~~ at least one of said computers in said **[[**P2P**]]** network; and

(C) obtaining said particular segment from said at least one of a plurality of computers in said **[[**P2P**]]** network of computers~~; and~~

~~(D) obtaining at least one other segment of said plurality of segments from at least one other computer in said P2P network~~.

(Amend. with Request for Continued Examination, Oct. 14, 2010, at 13-14; Ex. 1030.) The claims were subsequently allowed, without any further discussion of Woodhill.[7]

V. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. There is Nothing New About Segmenting Data Items and Using Segment Identifiers that are Functions (e.g., Hashes) of the Contents of These Segments

The '539 claims focus on the basic idea of using substantially unique

[7] Claim 17 during prosecution was eventually renumbered to challenged claim 21 and likewise contained many substantial amendments during prosecution. Claim 31 (introduced with the RCE) was renumbered to challenged claim 34.

identifiers – based on the contents of the segments contained within a data item – to access the data item. Claim 10, for example, recites:

> 10. A computer-implemented method of obtaining access to a data item at a first computer in a network of computers, said ***data item comprising a plurality of segments***, the method comprising the steps of:
>
> (A) in response to a request, said request comprising a first identifier, obtaining a plurality of segment identifiers, each of said segment identifiers corresponding to one of said plurality of segments, the ***segment identifier for each particular segment being based, at least in part, on a first given function of the data comprising said particular segment and only the data in said particular segment***, where any two identical segments will have identical segment identifiers as determined using said first given function wherein said ***first identifier is based at least in part, on a second given function of the plurality of segment identifiers***;
>
> (B) by hardware in combination with software, ***using at least one of said segment identifiers obtained in step (A), requesting at least one particular segment*** of said plurality of segments that comprise said data item from at least one of said computers in said network; and
>
> (C) ***obtaining said particular segment*** from said at least one of a plurality of computers in said network of

computers.

('539 patent, col. 41, ll. 57 – col. 42, ll. 12 (emphasis added); Ex. 1001.) The method involves using "segment identifiers" based at least in part on a "function of the data comprising [the] particular segment and only the data in [the] particular segment" (*e.g.*, a hash) and a "first identifier" based at least in part on "a second given function of the plurality of segment identifiers" (*e.g.,* a "hash of hashes"). (*Id*.)

The applicants stated that they were entitled to these broad claims because "[i]n ***all*** of the prior data processing systems the names or identifiers provided to identify data items . . . are ***always*** defined relative to a specific context," and "there is ***no direct relationship*** between the data names and the data item." (*Id*. at col. 2, ll. 26-31, ll. 39-40, emphasis added.)

These representations were simply wrong. Prior data processing systems ***did use*** identifiers based on the contents of a data item or its segments – and not the context or pathname – ***including*** identifiers based on a "hash of hashes." In fact, these techniques were old and widely used. This is not surprising. The concept of using a mathematical function to create a "fingerprint" or "signature" for a data item based on the content of the data item predates the '539 patent by decades. For example, IBM developed one of the first hash tables in the 1950s (*see*, *e.g.*, D. Knott, "Hashing functions", 18 The Computer Journal 265, 274

(1975) (discussing "history of hashing"); Ex. 1011), and Professor Ron Rivest of MIT introduced the MD5 hash algorithm referenced in the '539 patent in the early 1990s. (*See, e.g.*, R. Rivest, "The MD5 Message-Digest Algorithm," Internet RFC 1321 (Apr. 1992); Ex. 1012.)

Moreover, Professor Ralph Merkle and others were partitioning data items into segments, calculating identifiers for the segments using a hash function, and then "hashing the hashes" to create top-level signatures (*i.e.*, identifiers for the data items as a whole) by the 1970s.[8] (*See* U.S. Patent No 4,309,569 to Merkle, entitled "Method of Providing Digital Signatures," filed Sept. 5, 1979, at col. 2, ll. 54-67 and Figure 1 (describes calculating signatures for a "vector of data items" by calculating signatures for segmented portions of the vector using a hash function, then combining the signatures using the same hash function) ("Merkle"); Ex.

[8] The idea of partitioning data into smaller segments (e.g., "pages" or "blocks") has been known for decades. (*See, e.g.*, B. Lampson and R. Sproull, "An Open Operation System for a Single-User Machine," ACM Operating System Review (Dec. 1979) at 100 ("The system organizes long-term storage (on disk) into files, each of which is a sequence of fixed-size pages" and "[t]he data bytes of the file are contained in pages 1 through n."); Ex. 1032; *see also* A. Tanenbaum, "Operating Systems Design and Implementation," Prentice-Hall (1987) at 256; Ex. 1033.)

1031.) "Hash trees" – also known as "Merkle trees" – were well known in the field long before the '539 patent.

These hashing functions take as input the data contained in a file, segment, or other data item, and produce a much smaller-sized output value, commonly called a "hash," "hash value," "message digest" ("MD"), or "checksum." (*See, e.g.*, McGraw-Hill Dictionary of Scientific and Technical Terms, (4th ed., 1989), at 860; Ex. 1013; *see also* B. Kaliski, "A Survey of Encryption Standards," IEEE Micro (Dec. 1993), pp. 74-81, at 77; Ex. 1014.) For example, a file that is a million bytes (or even much larger) in size can be used as input to produce a hash value that is a mere 16 bytes in length. Because of the mathematical properties of the function, the odds that two different files will produce the same 16 byte hash are extremely small: for example, with a 16 byte hash output, the odds that two randomly picked inputs have the same hash are 2^{-64}, or approximately one in sixteen billion billions. (B. Kaliski at 77; Ex. 1014.) Consequently, hashes are known as "signatures" or "fingerprints" because they identify data items with high reliability, just like signatures or fingerprints identify people with a high degree of certainty. (*See* McGregor and Mariani, "'Fingerprinting' – A Technique for File Identification and Maintenance", 12 Software Practice & Experience 1165 (1982), at 1165 ("fingerprinting" technique "produce[s] a quasi-unique identifier for a file, derived from that file's contents . . . [t]he idea is to provide an identifying feature

for every file, which is intrinsically distinctive . . ."); Ex. 1017.)[9]

Although the applicants suggested in their patent application that they were the first to utilize hash functions to identify data items for file management applications, others working in the field used them for the same purposes more than a decade before the '539 patent. For example, at least sixteen years before the '539 patent was filed, researchers were already using hash functions to determine whether two records were identical, and to eliminate duplicate records. (*See*, *e.g.*, Babb, "Implementing a Relational Database by Means of Specialized Hardware", 4 ACM Transactions on Database Systems 1, at 2-4, (March 1979); Ex. 1034; Bitton and DeWitt, "Duplicate Record Elimination in Large Data Files", 8 ACM Transactions on Database Systems 255 at 256; Ex. 1035; *see also* Rabin,

---

[9] This reference was central to the rejection of EP counterpart application EP0826181A1 with claims having a central feature of content-based identifiers. (Annex to the communication, May 8, 2009; Ex. 1020.) The applicants amended the claims to emphasize a "licensing" limitation not found in the challenged claims (Reply to communication from the Examining Division, November 18, 2009 at 4; Ex. 1021.), but this too was found unpersuasive and the rejection was maintained by the EPO. (Annex to the communication, March 14, 2012 at 4; Ex. 1022) Following this rejection, Applicants withdrew the application from consideration. (Closing of Application, June 14, 2012; Ex. 1023.)

"Fingerprinting by Random Polynomials", Center for Research in Computing Technology, Harvard University, Report TR-15-81 at 1 and 9 (1981); Ex. 1015; Manber, "Finding Similar Files in a Large File System", Department of Computer Science, University of Arizona, Report TR 93-33 at 3 (1993); Ex. 1016; McGregor and Mariani, "'Fingerprinting' – A Technique for File Identification and Maintenance", 12 Software Prac. & Exp. 1165 (1982), at 1165; Ex. 1017.)

Many other printed publications and patents disclose and use identifiers exactly like those described and claimed in the '539 patent, including "hashes of hashes," for exactly the same purposes. These publications disclose identifiers that are location- and context-independent, that are determined using only the contents of a data item or a segment of a data item, and that are formed using identical algorithms to those mentioned in the '539 patent.

Browne: For example, researchers at the University of Tennessee and Bell Laboratories disclosed a system that created "location-independent file names" (or "LIFNs") to identify files on the Internet. (Browne at 3; Ex. 1002). LIFNs – like the identifiers in the '539 patent – uniquely identified files based on their contents, not their locations. (*Id*.; *compare* '539 patent, col. 35, ll. 39-41 (True Names used to identify files "independent of their present location"); Ex. 1001.) LIFN <signatures> were computed as "the ascii form of the MD5 signature of the file" – the same function identified in the '539 patent. (Browne at 6; Ex. 1002; *compare*

'539 patent, col. 14, ll. 51-52 (using MD5 or SHA); Ex. 1001.)

Browne specifically addressed compound data items ("resources"), including multiple files meant to be used together, such as a software package of computer program files. (Browne at 2, 5-6; Ex. 1002; *compare* '539, col. 41 ll. 49-56 (referencing "software products"); Ex. 1001.) To handle these compound resources, each component (*e.g.*, each file) of the resource was assigned its own LIFN <signature>, computed as the MD5 hash of the contents of the component. (Browne at 5-6; Ex. 1002.) The LIFNs for the individual components were then grouped together in a sequence, and a LIFN <signature> was computed for the resource as a whole by performing an MD5 hash of the sequence of LIFNs for the individual components. (*Id*. at 6.) In other words, hashes were computed for each component file, each hash acting as an identifier for its associated file, and a "hash of hashes" was then computed for the package, acting as the identifier for the package as a whole.

To access a resource stored on the network, a client would first use the LIFN for the resource (*e.g.*, the software package) to obtain the LIFNs for the individual components of the resource (*e.g.*, the individual files), and would then use the LIFNs for the individual components to retrieve those components from remote file servers on the network. (*Id*. at 4–6.) In each case, the LIFN <signature> would identify the file or resource. (*Id*. at 6.)

Langer: Another researcher, Albert Langer, also addressed the same problem as the '539 patent and, like Browne, proposed essentially the same solution. (Langer; Ex. 1003.) Langer was particularly concerned with sharing content on the Internet prior to the rise of the World Wide Web, through the use of popular protocols such as the File Transfer Protocol (FTP). FTP sites, among other things, could be accessed to provide a listing of available files at the site, and so a user could select and download files from the site. (*See, e.g.*, P. Deutsch et al., "How to Use Anonymous FTP," Internet RFC 1635 at 2 (May 1994); Ex. 1041.) Langer specifically addressed the problem of "uniquely identifying files which may have different names and/or be in different directories on different systems," and like the '539 applicants, observed that traditional location-based identifiers do not work well for distributed systems. (Langer at 3; Ex. 1003; *compare* '539 patent, col. 2, ll. 44-53; Ex. 1001.) Langer's solution, like the '539 patent, was to "provide a unique identifier for each file which is independent of location." (Langer at 3; Ex. 1003; *compare* '539 patent, col. 3, ll. 52-58; Ex. 1001.) Specifically, Langer disclosed "defining a unique identifier that does NOT include a particular site identifier," by "using a cryptographic hash function such as MD5," *i.e.*, the identical algorithm used in the '539 patent. (Langer at 4; Ex. 1003; *compare* '539 patent, col. 14, ll. 51-53; Ex. 1001.)

Langer, like Browne, also addressed the issue of compound data items,

including, for example, archived files that were part of the same package. (Langer at 5; Ex. 1003.) Langer extracted each file from the archive, and computed an identifier for it based on an MD5 hash of the contents of the file. (*Id.*) He then concatenated those identifiers together to create a new file (*i.e.*, a file of the sequence of MD5 hashes), and computed an MD5 hash of the contents of the new file (*i.e.*, a "hash of hashes") to serve as an identifier for the package as a whole. (*Id.*) A client could use the identifier for the package (*i.e.*, the hash of hashes) to obtain the file containing the sequence of MD5 hashes for the individual files, and then could select any of the MD5 hashes of the individual files to retrieve a particular item of interest from remote FTP sites on the network. (*Id.* at 3, 5.)

Kantor: Dr. Frederick W. Kantor, a physicist from Columbia University, developed yet another example of context- and location-independent identifiers for the same purposes as the '539 patent. Dr. Kantor described a product called FWKCS that created "contents_signatures" for files based on their content. (Kantor at Preface 2; Ex. 1004.)[10] FWKCS used these contents_signatures to uniquely identify files on a bulletin board system ("BBS"), an online file system

[10] The three-page Preface section of Kantor's FWKCS user manual does not have individual page numbers. Citations to the Preface are labeled "Preface" to denote pages 1-3 of the Preface section. Otherwise, citations refer to the page numbers in the top-right margin of the remainder of the user manual.

considered a precursor to the World Wide Web. (*Id.* at Preface 2.) The contents_signature, as the name suggests, was based on a hash of the data contained in a file, just like the identifiers in the '539 patent. (*Id.* at 8; *compare* '539 patent, col. 15, ll. 37-48 and Figure 10A; Ex. 1001.)

Kantor, like Browne and Langer, also specifically addressed the issue of compound data items, in particular, "zipfiles" containing a set of other files meant to be kept together. (Kantor at Preface 1, Preface 2; Ex. 1004.) FWKCS created "zipfile contents signature[s]" for these zipfiles, based on a hash of the contents_signatures of the files within the zipfile (*i.e.*, a "hash of hashes"), once again, just as in the '539 patent. (Kantor at 9; Ex. 1004; *compare* '539 patent, col. 15, ll. 49-67 and Figure 10B; Ex. 1001.) FWKCS's contents_signatures accordingly could be used both to identify compound data items, like zipfiles, and to separately identify the segments (*i.e.*, individual files) contained within them. (Kantor at Preface 2; Ex. 1004.)

Woodhill: The Woodhill patent provides still another example of the use of context- and location-independent identifiers for the same purposes as the '539 patent. Woodhill created a distributed storage system that used "Binary Object Identifiers" to identify, access, and back-up files, among other functions. (Woodhill at Abstract; Ex. 1005) As Woodhill explains, these Binary Object Identifiers provided "a unique identifier for each binary object to be backed up."

(*Id.* at col. 4, ll. 45-47, Ex. 1005). The Binary Object Identifiers included three fields – a CRC value, a LRC value, and a hash value – "[e]ach . . . calculated from the contents of each binary object." (*Id*. at col. 7, ll. 64 – col. 8, ll. 32).

For larger files, the binary objects could be further "granularized," that is, divided into smaller segments of data called "granules." (*Id*. at col. 15, ll. 9-24.) Woodhill computed a "contents identifier" for each of the granules based on a CRC value and hash value "calculated against the contents of the 'granule.'" (*Id*. at col. 15, ll. 24-28.) These granule identifiers were used to identify the corresponding data for a variety of purposes, including "backup" and "restore" operations. For example, to restore a file to a former version, the granule identifiers could be used to request and obtain from a remote server only those particular granules that were different between the former and the current versions of the file, thus reducing the amount of data that needed to be transmitted to restore the file. (*Id*. at col. 17, ll. 18-27.)

These prior art references provide just a handful of many examples of the use of content-based identifiers, including "hashes of hashes," to perform basic file management functions. Indeed, the application of hash-based identifiers to these functions was so obvious that at least one commentator not only described the applications as "easy" but also posted these ideas publicly "to impede anyone who might independently have had the idea from patenting it." (Williams, "An

algorithm for matching text (possibly original)", posted to the "comp.compression" newsgroup on January 27, 1992 at 1; Ex. 1037; *see also* R. Williams, "An Introduction to Digest Algorithms," Rocksoft (Nov. 1994), at 11-13 (further describing potential uses for file management purposes of identifiers based on the hash of the contents of a block of data); Ex. 1038.)

In short, other than perhaps coining a new phrase – i.e., True Name – for a very old concept, there is absolutely nothing new disclosed or claimed in the '539 patent concerning the use of location-independent, content-based data identifiers.

VI. SPECIFIC GROUNDS FOR PETITION

Pursuant to Rule 42.104(b)(4)-(5) and Practice Guide Fed. Register Vol. 77, No. 27, page 6873 Petitioners have submitted claim charts in connection with this Petition (attached as Exs. 1045-1048), which have been submitted in the pending litigation between Petitioner and PersonalWeb Technologies LLC. Those charts support Petitioners' position with respect to those references and demonstrate that the challenged claims are anticipated and/or unpatentable in view of each of them. Petitioner also submits herewith the Declaration of Dr. Douglas Clark (Ex. 1009), a Professor of Computer Science at Princeton University. Dr. Clark confirms that the charts identify representative subject matter in each reference that teaches each and every limitation of the challenged claims. He likewise confirms how each claim is anticipated or, at a minimum, rendered obvious by the prior art.

A. Grounds of Invalidity for Challenged Claims 10, 21, and 34 based on Browne as a Primary Reference

Ground 1: Browne Anticipates Challenged Claims 10, 21 and 34

Browne was not referenced or discussed by the examiner during prosecution of the '539 patent.[11] It is prior art under at least 35 U.S.C. § 102(a) and anticipates each of claims 10, 21 and 34 of the '539 patent.

Browne describes the Bulk File Distribution ("BFD") package developed by researchers at the University of Tennessee and Bell Laboratories as part of an effort to make scientific software easily accessible over the Internet. (Browne at 1, 6; Ex. 1002.) The BFD package is based on the concept of a "virtual repository," which is a distributed network of physical software repositories, each residing on a different file server. (*Id*. at 1-2.)

Like the '539 patent, Browne begins by discussing the shortcomings of context- or location-dependent file identifiers. At the time, a virtual repository could be implemented using a Uniform Resource Locator (URL) to identify each file. (*Id*. at 2.)[12] The authors identify several problems with the use of location-

---

[11] Browne is cited on the face of the '539 patent as one of the over 400 references. Browne played no role in the prosecution of the '539 patent.

[12] A URL is a character string, such as "http://www.netlib.org/index.html," that can be used to specify a transfer protocol ("HTTP"), a location ("www.netlib.org"),

based identifiers, such as URLs, to access virtual software repositories. Among other things, URLs are inadequate for ensuring the consistency of a software repository. (*Id*. at 2.) Moreover, a URL can only identify a single location; if a virtual repository offers multiple copies of the same file, each copy must be given its own URL. (*Id*. at 2.)

In order to address these shortcomings, Browne adopts the same solution that would be later proposed in the '539 patent: associating a unique identifier with the ***contents*** of a file, rather than with the ***location*** of the file. Indeed, Browne even uses the same terminology as the patent, referring to its file names as "location independent." In the BFD package, the identifier is called a "Location Independent File Name," or LIFN. (Browne at 3; Ex. 1002; *compare* '539 patent, col. 3, ll. 55-58 ("the identity of a data item is ***independent*** of its name, origin, ***location*** . . . .")(emphasis added).)

In Browne's preferred approach, the LIFN is computed as the MD5 hash of the contents of a file. (Browne at 6; Ex. 1002.) The MD5 algorithm provides a substantially unique fingerprint, meaning that two files with identical content will always have the same MD5 fingerprint, even if they are located on different

---

and a file name ("index.html"). (*See, e.g*., T. Berners-Lee et al., "Uniform Resource Locators (URL)," Internet RFC 1738 (Dec. 1994) at 1; Ex. 1018.)

servers, and even if the server administrators give them different names.[13] "Once a LIFN has been assigned to a particular sequence of bytes, that binding may not be changed." (*Id*. at 3.)

To access a file, a client computer sends a query to a LIFN server including the LIFN <signature> (*i.e.,* the MD5 hash) of the desired file to be accessed. *(Id.* at 4-5.) In response, the LIFN server returns a list of file servers that store a copy of the file associated with that LIFN <signature>. (*Id*. at 4-5.) To be clear, this mechanism is just like the True File Registry (TFR) of the '539 patent, which receives a True Name identifier and provides a list of file servers (source IDs) that store a copy of that file. ('539 patent, col. 35, ll. 47–49; Ex. 1001.)

Browne also addresses compound data items (*e.g*., related files meant to be used together) in the same manner as the '539 patent. He refers to these compound items as "resources," and specifically addresses the need to ensure consistency between them. *(Id.* at 2, 5-6; Ex. 1002.) To achieve that goal, Browne computes LIFNs for each of the components (*e.g*., files) that make up the resource, based on an MD5 hash of the component. (*Id*. at 6.) The identifiers are then combined in a

---

[13] The general syntax for the LIFN is "lifn:netlib:<signature>", referencing the file access protocol ("lifn," similar to the "http" protocol identifier in a URL), the server handling the request ("netlib"), and the unique MD5 hash used to identify a file[13] ("<signature>"). (Browne at 4, 6; Ex. 1002.)

sequence to obtain a new file, called a "composite-parts-list," and a LIFN is computed for the composite-parts-list by performing an MD5 hash of the sequence of LIFNs for the individual components (*i.e.*, a "hash of hashes"). (*Id*. at 6.)

To access a particular component of a compound resource (*e.g.*, a particular file), a client sends the LIFN <signature> of the compound resource to a LIFN server, and in response, obtains a list of the locations where it can obtain the composite-part-list file. (*Id*. at 5-6.) The client then downloads the composite-part-list file from one of those locations, and uses the LIFN <signatures> contained in the file to retrieve the desired file. (*Id*. at 5-6.)

As set forth in detail in the claim chart (Ex. 1045), and as confirmed by Dr. Clark (Clark Declaration at ¶¶ 17-21; Ex. 1009), Browne anticipates each of claims 10, 21 and 34 of the '539 patent. For example, claim 10 (which is similar to claim 21) recites:

> 10. A computer-implemented method of obtaining access to a data item at a first computer in a network of computers, said data item comprising a plurality of segments, the method comprising the steps of:
>
> (A) in response to a request, said request comprising a first identifier, obtaining a plurality of segment identifiers, each of said segment identifiers corresponding to one of said plurality of segments, the segment identifier for each particular segment being

> based, at least in part, on a first given function of the data comprising said particular segment and only the data in said particular segment, where any two identical segments will have identical segment identifiers as determined using said first given function, wherein said first identifier is based, at least in part, on a second given function of the plurality of segment identifiers;
> (B) by hardware in combination with software, using at least one of said segment identifiers obtained in step (A), requesting at least one particular segment of said plurality of segments that comprise said data item from at least one of said computers in said network; and
> (C) obtaining said particular segment from said at least one of a plurality of computers in said network of computers.

('539 patent, col. 41, l. 57 – col. 42, l. 12; Ex. 1001.)

Regarding the limitation (A) in claims 10 and 21, Dr. Clark confirms that Brown discloses a method that, in response to a request including a "first identifier" (the LIFN <signature> of the resource), obtains the "segment identifiers" for each of the components or segments comprising the data item (the LIFN <signatures> of the individual files). (Clark Decl., ¶ 19; Ex. 1009.) Dr. Clark further confirms that the function used to obtain the "segment identifiers" (the MD5 hash) is "based, at least in part, on a function of the data" comprising the

segments (MD5 is based on the data), that this function ensures that "any two identical segments will have identical segment identifiers" (this is a property of MD5), and that "first identifier is based, at least in part, on a second given function of the plurality of segment identifiers" (the LIFN <signature> for the compound resource is the MD5 hash of the LIFNs for the individual files contained in the resource). (Clark Decl., ¶ ; Ex19; Ex. 1009.)

Regarding limitations (B) and (C) in claim 10 and limitations (B), (b1), and (b2) in claim 21, Dr. Clark confirms that Browne discloses a method that uses "hardware in combination with software" (a software program), and that uses one of the "segment identifiers" (LIFN <signatures> for a file in a resource) to request and obtain a copy of a segment (the identified file in the resource) from one of the computers in the network (one of the servers in the network storing files). (Clark Decl., ¶ 20; Ex. 1009.) Clark confirms that Brown discloses "using a particular segment identifier. . . particular segment" in (b0) of claim 21. (Clark Decl., ¶ 21; Ex. 1009; Browne at 5-6; Ex. 1002.)

Claim 34 recites "dividing" a particular data item into a plurality of segments, as well as determining "segment identifiers" being a "True Name" for each segment, and a data identifier for the data item as a while being a "True Name" of the segment identifiers. Dr. Clark confirms that Browne discloses the limitations "(A) dividing . . . " (Clark Decl., ¶ 19; Ex. 1009; Browne at 5-6; Ex.

1002) , "(B) determining . . . segment", (Clark Decl., ¶ 19; Ex. 1009; Browne at 5-6; Ex. 1002), "(C) forming . . . identifiers" (Clark Decl., ¶ 19; Ex. 1009; Browne at 5-6; Ex. 1002), "(D) determining . . . item, and" (Clark Decl., ¶ 19; Ex. 1009; Browne at 5-6; Ex. 1002), "(E) maintaining . . . item" (Clark Decl., ¶ 19; Ex. 1009; Browne at 5-6; Ex. 1002) and "(F) in response. . . item." (Clark Decl., ¶ 20; Ex. 1009; Browne at 4-5; Ex. 1002)

Ground 2: Challenged Claim 34 is Unpatentable as Obvious in view of Browne in Combination with Woodhill

In the event PersonalWeb contends that Browne does not satisfy the claim limitations of "dividing a particular data item into a plurality of segments" (claim 34), or "segments" or "segment identifiers" (all claims) a person of ordinary skill would have found it obvious to modify Browne to divide data items into segments and to create identifiers for those segments. As Dr. Clark confirms, dividing a large file into segments was a well-known technique to handle large files, such as databases. (Clark Decl., ¶ 23; Ex. 1009.) For example, U.S. Patent No. 5,649,196 to Woodhill et al. ("Woodhill") discloses dividing a data item into a plurality of segments (e.g., dividing files into "binary objects," and further dividing the binary object into "granules").[14] (Woodhill at col. 15, ll. 11-38, col. 22, ll. 51-54, col. 24 ll. 18-19; Ex. 1005.) Woodhill teaches that dividing files into smaller segments

[14] See also footnote 8 *supra*.

(e.g., "binary objects," and "granules") is a known and effective technique to reduce the amount of data that must be transmitted (i.e., smaller segments instead of entire files are transmitted). (Clark Decl., ¶ 23; Ex. 1009; Woodhill at col. 15, ll. 4-6; Ex. 1005.)

Similarly, in the event PersonalWeb contends that Browne does not satisfy the claim limitation of a "TrueName" (claim 34) because the LIFN <signatures> do not include an indication of length, a person of ordinary skill in the art would have found it obvious to modify Browne to append an indication of length to the LIFN <signatures>. For example, Woodhill includes a length value in its Binary Object Identifiers. (Woodhill at col. 8, ll. 4-5; Ex. 1005.) Dr. Clark confirms that applying Woodhill's indication of length to Browne would have been obvious to a person of ordinary skill in the art.[15] (Clark Decl., ¶ 25; Ex. 1009.)

B. Grounds of Invalidity for Challenged Claims 10, 21, and 34 based on Langer as a Primary Reference

Ground 3: Langer Anticipates Challenged Claims 10, 21 and 34

Langer was not referenced or discussed by the examiner during prosecution of the '539 patent.[16] It is prior art under at least 35 U.S.C. § 102(b) and anticipates

---

[15] The "dividing . . ." step could also be obtained by combining Browne with other prior art references, such as Kantor. (*See, e.g.*, Clark Decl., ¶¶ 23, 25; Ex. 1009.)

[16] Like Browne, Langer is cited on the face of the '539 patent as one of over 400

each of claims 10, 21 and 34 of the '539 patent.

Langer addresses the problem of distributing files over the Internet. Langer predates the advent of the World Wide Web, and therefore focuses on earlier file distribution technologies, notably the *File Transfer Protocol* (FTP) and the *Archie* and *WAIS* search engines. (Langer at 2; Ex. 1003.) Langer provided his contribution to the "alt.sources.d" and "comp.archives.admin" Usenet newsgroups. At the time, Usenet was one of the most effective channels for researchers to discuss current technical issues and to distribute research materials.

Like Browne, Langer recognizes the limitations inherent in the use of context- or location-based file identifiers, and the benefits of "uniquely identifying files which may have different names and/or be in different directories on different systems." (*Id*.at 3.) For example, identifiers that are tied to a physical server do not allow a user to select another site that is physically closer. (*Id*. at 3.) Langer's solution is exactly the same as the '539 patent: determine a substantially unique identifier for each file based on the ***content*** of the file rather than its ***location***, and associate that file with the unique identifier (*Id*. at 3-4; Ex. 1003; *compare* '539 patent, col. 3, ll. 55-58; Ex. 1001.) Langer expressly recognizes that such an identifier may be calculated by performing a hash function on the contents of the

---

references. Langer played no role in the prosecution of the '539 patent.

file:

> A simple method of defining a unique identifier that does NOT include a particular site identifier would be to use a hash function on the entire contents of the file. . . . I would suggest using a cryptographic hash function such as MD5 which generates a 16 byte result.

(Langer at 4; Ex. 1003.) The'539 patent tracks Langer's solution (which predates it by almost four years) down to the choice of the same MD5 hash function.

Like Browne, Langer also specifically addresses compound data items, including, for example, archived files that are part of the same package. (*Id*. at 5.) Langer observes that such a package may be distributed in a variety of archive formats, and thus files may appear to be different even though they have identical content. (*Id*.) To address these compound data items, Langer divides the packages into their component files, and computes a unique identifier for each component by performing an MD5 hash on the contents of the component. (*Id*.) He then concatenates these identifiers for the components together in a sequence to create a new file (*i.e.*, a file of the sequence of MD5 hashes), and performs another MD5 hash on the contents of the new file (*i.e.*, a "hash of hashes") to serve as an identifier for the package as a whole. (*Id*.) Once again, this is the same algorithm adopted years later by the '539 patent to compute True Names for "compound data items." ('539 patent, col. 15, ll. 37-67 and Fig. 10(b); Ex. 1001.)

Langer uses these substantially unique identifiers with a central database server, such as the Archie and WAIS search engines, that associates MD5 hashes with physical locations. (Langer at 3-4; Ex. 1003.) Based on this infrastructure, a client computer can access a file using its identifier. For example, a user can query the Archie or WAIS search engines to find which FTP server holds a copy of a file with a specified MD5 hash. (*Id*. at 3-4; Ex. 1003; *see also, e.g.*, EARN Staff, "Guide to Network Resource Tools," Internet RFC 1580 (March 1994) at 23-26 (WAIS), 29-31, 36-37 (Archie); Ex. 1039.) The client computer can then access the file from one of the previously-identified file servers, again using the file's unique identifier.[17] Langer's central database server thus directly anticipates the True File Registry of the '539 patent, which provides a list of the locations, such as file servers, where a file with a given True Name is stored. ('539 patent, col. 35, ll. 47-49; Ex. 1001.)

The same mechanism is used to access compound data items, such as archived packages. (Langer at 5; Ex. 1003.) A client computer can use the MD5

[17] Similarly to Browne, Langer proposes to alias MD5 signatures to actual file names on the server: "A simple ftp implementation would just hardlink every file available for ftp to a filename encoding of it's [sic] MD5 token. Users would then ftp the directory path and filename of the MD5 token and obtain the file." (Langer at 4; Ex. 1003.)

identifier for the package (i.e., the "hash of hashes") to obtain the sequence of MD5 identifiers for the individual files, and then can use the MD5 identifiers for the component files to retrieve any particular file from remote FTP sites on the network. (*Id*. at 3-5.)

As set forth in detail in the claim chart (Ex. 1046), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 26-29; Ex. 1009), Langer anticipates each of claims 10, 21 and 34 of the '539 patent. For example, for claims 10 and 21, with regard to limitation (A), Dr. Clark confirms that Langer discloses a method that, in response to a request including a "first identifier" (the MD5 hash of the package), obtains the "segment identifiers" for each of the segments (the MD5 hashes for the files in the package). (Clark Decl., ¶ 29; Ex. 1009; Langer at 3-5; Ex. 1003.) Dr. Clark further confirms that the function used to obtain the "segment identifiers" (the MD5 hash) is "based, at least in part, on a function of the data" comprising the segments (the MD5 hash is calculated based on the data in the files), that this function ensures that "any two identical segments will have identical segment identifiers" (this is a property of MD5), and that the "first identifier" is based, at least in part, on a second function of the segment identifiers" (the identifier for the package is an MD5 hash of the identifiers for the component files). (Clark Decl., ¶ 28; Ex. 1009; Langer at 5; Ex. 1003.)

Regarding the limitations (B) and (C) in claim 10 and limitations (B), (b1),

and (b2) in claim 21, Dr. Clark explains that Langer discloses a method that uses "hardware in combination with software" (software running on an FTP file server), and that uses one of the segment identifiers (an MD5 hash of a file in the package) to request and obtain a copy of the segment (the file) from one of the computers in the network (one of the FTP sites). (Clark Decl., ¶ 29; Ex. 1009; Langer at 3-4; Ex. 1003.) Dr. Clark confirms that Langer discloses "using a particular segment identifier . . . particular segment" in (b0) of claim 21. (Clark Decl., ¶ 29; Ex. 1009; Langer at 4; Ex. 1003.)

Regarding claim 34, Clark confirms that Langer discloses the limitations "(A) dividing . . . segments" (Clark Decl., ¶ 28; Ex. 1009; Langer at 4-5; Ex. 1003.) , "(B) determining . . . segment", (Clark Decl., ¶ 28 ; Ex. 1009; Langer at 5; Ex. 1003.), "(C) forming . . . identifiers" (Clark Decl., ¶ 28; Ex. 1009; Langer at 5; Ex. 1003.), "(D) determining . . . item, and" (Clark Decl., ¶ 28; Ex. 1009; Langer at 5; Ex. 1003.), "(E) maintaining . . . item" (Clark Decl., ¶ 28; Ex. 1009; Langer at 5; Ex. 1003.) and "(F) in response. . . item." (Clark Decl., ¶ 29; Ex. 1009; Langer at 5; Ex. 1003.) Thus claim 34 is anticipated by Langer.

Ground 4: Challenged Claim 34 is Unpatentable as Obvious in view of Langer in combination with Woodhill

In the event PersonalWeb contends that Langer does not satisfy the claim limitation of "dividing a particular data item into a plurality of segments" (claim

34), or "segments" or "segment identifiers" (all claims) a person of ordinary skill would have found it obvious to modify Langer to meet that limitation, for substantially the same reasons outlined above in reference to the combination of Browne with Woodhill. As Dr. Clark explains, Woodhill teaches that dividing files into smaller segments (e.g., "binary objects," and "binary objects" into "granules") is a known effective technique to reduce the amount of data that must be transmitted (i.e., smaller segments instead of entire files are transmitted). (Clark Decl., ¶¶ 30-31; Ex. 1009; Woodhill at col. 15, ll. 13-20; Ex. 1005.)[18]

Similarly, in the event PersonalWeb contends that Langer does not satisfy the claim limitation of a "TrueName" (claim 34) because this requires a hash of the contents of a data item as well as a length indication, a person of ordinary skill would have found it obvious to modify Langer to append an indication of length to the MD5 identifiers. Woodhill includes a length value in its Binary Object Identifiers. Dr. Clark confirms that applying Woodhill's segmentation technique to Langer would have been obvious to a person of ordinary skill in the art. (Clark Decl., ¶¶30-31; Ex. 1009.) This analysis is set forth in detail in the claim chart.

C. Grounds of Invalidity for Challenged Claims 10, 21, and 34 based

[18] These limitations could also be shown by combining Langer with any of the other prior art references cited in this petition, such as Kantor. (*See, e.g.*, Clark Decl., ¶¶ 19, 43; Ex. 1009.)

on Kantor as a Primary Reference

Ground 5: Kantor Renders Challenged Claims 10, 21, and 34 as Obvious

Kantor was not cited to the USPTO and not considered during prosecution of the '539 patent. It is prior art under at least 35 U.S.C. § 102(b) and renders each of claims 10, 21, and 34 of the '539 patent obvious.

Kantor is a published manual that describes a software program called the Frederick W. Kantor Contents-Signature System Version 1.22 ("FWKCS"). (Kantor at Title Page; Ex. 1004.) Like the '539 patent, Kantor addresses the shortcomings of context- or location-dependent file identifiers. (Kantor at Preface 1; Ex. 1004; *compare* '539 patent, col. 3, ll. 30-44; Ex. 1001.) These include the "problem of duplicate files on electronic bulletin board systems" or BBSs.[19] (Kantor at Preface 1; Ex. 1004.) BBS users would unwittingly or intentionally upload files to a bulletin board, which the bulletin board already had. (*Id*.) Consequently, bulletin board operators "were paying for hardware to provide the capacity for these spurious [duplicate] files, and spending many hours trying to find and delete them." (*Id*.)

Kantor, like Browne and Langer, uses the same solution that would be later

[19] Before the World Wide Web, computers "dialed into" a file server or network of servers where users could exchange files or other information by uploading or downloading files.

proposed in the '539 patent: associating a unique identifier with the ***contents*** of a file, rather than with the ***location*** of the file. Kantor calls these identifiers "contents-signatures," and uses them to identify a file based only on the contents of a file, and not its name, location, or other characteristics. (*Id*.; *compare* '539 patent, col. 3, ll. 30-44; Ex. 1001.) These signatures can be used for various purposes, including, for example, identifying duplicate content already stored on the BBS system (*.e.g.*, Kantor at Preface 2-3; Ex. 1004), using a "Lookup" command to identify whether a file to be uploaded to a BBS is already present on the BBS (*id.* at 173), and using a "Precheck" command to generate a report identifying files which are present on the BBS system but not on the user's computer. (*Id*.)

FWKCS computes the contents-signature based on a function of the data in a file. (*See id.* at 7-8.) Specifically, the contents-signature is constructed with "the 32-bit CRC [cyclic redundancy check][20] of the file contents and the uncompressed file-length." (*Id*.) The CRC and the file length (*i.e.,* file size) are both a function of the data contained in the file, and two files with the same content necessarily have the same contents-signature. (*Id*.) In fact, Kantor uses the same technique as

[20] As Dr. Clark confirms, a CRC is a well-known hash function that calculates a value as a function of the file's contents. (Clark Decl., ¶ 34; Ex. 1009.)

in the '539 patent, creating a contents-signature with a hash and a length value. (Kantor at 7-8; Ex. 1004; *compare* '539 patent, col. 15, ll. 37-48 and Figure 10A; Ex. 1001.) Each contents-signature is location-independent and independent of the file's pathname, location, or context.

Kantor also creates identifiers for compound data items in the same manner as the '539 patent. Kantor explains that BBS users often bundled files into "zipfile" format, a well-known format for organizing related files into a single compound file. (*See* Kantor at Preface 2; Ex. 1004.) FWKCS generates "zipfile contents-signatures" for these zipfiles by computing the contents-signatures for each of the individual files within the zipfile, then hashing these contents-signatures using an "addition modulo 2^32" hash[21] to create the zipfile contents-

[21] As Dr. Clark confirms, "addition modulo 2^32" is another well-known hash function that uses addition to calculate a value based on a file's contents. (Clark Decl., ¶ 35; Ex. 1009; *see also* D. Knott, Hashing functions, The Computer Journal 18 (1975), vol. 3, at 268 (describing "common elementary hashing functions" including addition functions); Ex. 1011.) By adding together the values of the contents identifiers for the individual files, Kantor ensures that "the resulting [zipfile contents identifier] does not depend on the names of the files, the dates of the files, [or] the order in which they appear in the zipfile . . . ." (Kantor at 9; Ex. 1004.)

signature for the zipfile as a whole (*i.e.,* a "hash of hashes"). (*Id*. at 9)

FWKCS provides many operations for working with these zipfile and file contents-signatures. For example, FWKCS computes the zipfile and file contents-signatures for all of the zipfiles and individual files in the system, and stores them in a master contents-signature list, such as "CSLIST.SRT," which is similar to the "True Name Registry" of the '539 patent. (*Id.* at 18; *compare* '539 patent, col. 35, ll. 47-49; Ex. 1001). In addition, when uploading a zipfile, FWKCS can determine whether that zipfile already exists in the system using the zipfile contents-signature, and can determine whether individual component files of that zipfile already exist in the system, using the contents-signatures for the individual files. (Kantor at 9; Ex. 1004.) The zipfile and file contents-signatures also can be used to find zipfiles or files on the BBS, to delete duplicate zipfiles or files uploaded under different names, and to determine if files are contained in a larger zipfile or spread among different zipfiles. (*Id.* at 9).

Although BBS clients typically connected to a BBS and requested files based on the file's name (*See, e.g.*, F. Clark et al., "PCBoard v15.0 Technical Reference Manual," Clark Development Corporation, 1993 at 332[22]; Ex.1040 ; *see*

[22] Clark Development Corporation released PCBoard v15.0 in August 1993. Page numbers were added to the documentation included in this release for Ex. 1040 to

also Kantor at Preface 1; Ex. 1004), a person of ordinary skill in the art would have found it obvious to modify the BBS commands, including the download and/or read commands, to permit identifying files based on contents-signatures or zipfile contents-signatures. (Clark Decl., ¶ 41; Ex. 1009.) Among other things, this would facilitate integrity checking by more precisely specifying the file of interest by its content, and thus improve accuracy. (*Id*. at ¶ 41.) Kantor shows that such a modification would be easy to implement. For example, FWKCS already had utilized contents-signatures as parameters specified in certain user commands, such as the "Lookup" operation (*see* Kantor at 97 and 173; Ex. 1004.), and it would have been straightforward to similarly allow download and read commands to identify a file by a contents-signature. (Clark Decl., ¶ 41; Ex. 1009.) Moreover, it would be an easy matter for a user to obtain the contents-signatures for the files of interest. For example, the signatures could be shared among users. (*Id*. at ¶ 41.) In addition, the signatures could be provided to a user by the BBS itself using FWKCS through the Precheck operation or an easily modified version of this operation. (*Id*. at ¶¶ 39, 41.) Kantor describes the Precheck operation as an FWKCS utility for identifying files, based on their contents-signatures, which exist on the BBS but which do not yet reside on a user's computer. (Kantor at 173; Ex.

---

conform to 37 C.F.R. 42.63.

1004.) Using Precheck, a user is provided a report and can then use contents-signatures from the report to request files of interest with the modified download command. (Clark Decl., ¶¶ 39, 41; Ex. 1009.) In addition, Kantor provided contents-signatures to the user in response to Lookup commands in certain modes of operation. (Kantor at 96-97; Ex. 1004; Clark Decl., ¶37; Ex. 1009.) A person of ordinary skill in the art also would have found it obvious to divide a particular data item, such as a zipfile, into a plurality of segments. As Dr. Clark confirms, dividing a file into parts was a well-known technique to handle large files, such as databases, and is known and effective technique for reducing the amount of data that must be transmitted over a network. (Clark Decl., ¶ 31; Ex. 1009.) Kantor discloses "unzipping" a zip file to access the individual files within it. (Kantor at 174; Ex. 1004.)

As set forth in detail in the claim chart (Ex. 1047), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 41-43), Kantor renders obvious each of claims 10, 21 and 34 of the '539 patent. For example, for claims 10 and 21, regarding limitation (A), Dr. Clark confirms that Kantor could be easily modified to include a modified download or read command identifying zipfiles using a zipfile contents-signature, rather than using a pathname. In response to such a request, the modified BBS would obtain the "segment identifiers" for each of the segments contained in the zipfile (the contents-signatures for the individual files) and return them to the user.

(Clark Decl., ¶ 41; Ex. 1009; Kantor at 96-97; Ex.1004.) Dr. Clark further confirms that the "segment identifiers" are obtained by a function (the CRC hash) that is "based, at least in part, on a function of the data" comprising the segments, and that this hash function ensures that "any two identical segments will have identical segment identifiers." (Clark Decl., ¶ 34; Ex. 1009; Kantor at 10-11; Ex. 1004.) The "first identifier" (the zipfile contents signature) is based at least in part on a "second given function of the plurality of segment identifiers" because it is based on an addition modulo 2^32 hash of the contents-identifiers for the individual files within the zipfile. (Clark Decl., ¶ 35 ; Ex. 1009; Kantor at 52-54; Ex. 1004.)

Regarding the limitations (B) and (C) in claim 10 and limitations (B), (b1), and (b2) in claim 21, Dr. Clark confirms that the Kantor discloses using "hardware in combination with software" (the FWKCS and BBS software and hardware), and that, with the above-described obvious modification to Kantor, a "segment identifier" (a contents-signature for an individual file within a zipfile) could be used in a modified download command to "request" (through the modified download command) and obtain a segment (the file) from one of the computers on the BBS. (Clark Decl., ¶ 41; Ex. 1009, Kantor at 96-97; Ex. 1004.) Dr. Clark confirms that Kantor discloses "using a particular segment identifier . . . particular segment" in (b0) of claim 21. (Clark Decl., ¶ 36; Ex. 1009, Kantor at 52-54; Ex.

1004.) Regarding claim 34, Clark confirms that Kantor renders obvious all limitations of the claim, including "(A) dividing . . . segments" (Kantor at 174; Ex. 1004)," "(B) determining . . . segment", (Kantor at 7, 8, 48, 51, 55; Ex. 1002), "(C) forming . . . identifiers" (Kantor at 9; Ex. 1004), "(D) determining . . . item, and" (Kantor at 9; Ex. 1004), "(E) maintaining . . . item" (Kantor at 53; Ex. 1004) , and and "(F) in response. . . item." (Kantor at 96; Ex. 1004) . (*See* Clark Decl., ¶ 41; Ex. 1009.) With the above-described obvious modification to Kantor, a "data item identifier" (a zipfile contents signature could be used in a modified download command to "request" (through the modified download command) and obtain a data item (the zipfile) from one of the computers on the BBS. (Clark Decl., ¶ 41; Ex. 1009.)

Grounds 6 and 7: Kantor in view of Browne (Ground 6) or Langer (Ground 7) Renders Claims 34 as Obvious

To the extent PersonalWeb contends that "a True Name of the data" is limited to use of a cryptographic hash function such as MD5, a person of ordinary skill in the art would have found it obvious to modify Kantor to utilize an MD5 hash. Modifying Kantor's contents-signature structure to use MD5 instead of CRC would have been desirable to improve the statistical error rate of FWKCS further using a hash function widely accepted in the community. The use of such an MD5 hash for the same purposes is disclosed both in Browne (*see* Browne at 2-9; Ex.

1002 (using MD5 as part of the LIFN signatures)) and Langer (*see* Langer at 2-5; Ex. 1003). The modification of the contents-signature format to use MD5 instead of CRC would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results, and therefore it would be obvious to a person of ordinary skill in the art. (Clark Decl., ¶¶ 45-46; Ex. 1009.) Further, design incentives and market forces would have prompted the application of any of the LIFN Prior Art to Kantor, because the resulting system would have provided for collision-resistant generation of identifiers using a known function, and further would have simplified the generation of identifiers for a file that includes other files. (*Id*. at ¶¶ 42-43.) Finally, the combination would have required no more than ordinary creativity, taking into account the inferences and creative steps that a person of ordinary skill in the art would employ. (*Id*. at ¶¶ 42-43.)

D. Grounds of Invalidity for Challenged Claims 10, 21 and 34 based on Woodhill as a Primary Reference

Ground 8: Challenged Claims 10 and 21are Unpatentable as Obvious in view of Woodhill in combination with Fischer

Woodhill was considered during prosecution of the ‘539 patent, but was considered alone (not in combination with other art) and for claims substantially

different to those now being challenged.[23] Woodhill is prior art under at least 35 U.S.C. § 102(e) and, when considered in combination with Fischer, renders obvious each of claims 10, 21, and 34 of the '539 patent.[24]

Woodhill discloses a distributed storage management system that, as its title suggests, includes mechanisms for backing up, restoring, and accessing the files stored by each computer in the system. (Woodhill at col. 2, ll. 39-49; Ex. 1005.) Woodhill views these files "as a collection of data streams," each of which is a "distinct collection of data within the file that may be changed independently from other distinct collections of data within the file." (*Id*. at col. 4, ll. 14-18.) The

---

[23] The claims considered in light of Woodhill alone lacked, for example, the content-based "segment identifiers" now at the core of the asserted claims. There is no indication in the file history that the Patent Office considered Woodhill's granularization technique, or the associated technique for restoring previous versions of granularized binary objects, now relevant to the challenged claims.

[24] In the pending case of U.S. App. No. 13/352,169, family member of the '539 patent with claims substantially similar to claims 10, 21, and 34 of the '539 patent, claims have been rejected (over applicants' attempts to distinguish) based on portions of Woodhill that the present Petition relies on, and which the prosecution of the '539 did not rely on. (*See* Non-Final Office Action, March 27, 2012, at 5; Ex 1042; *see also* Response to Non-Final Office Action, June 26, 2012, at 22; Ex. 1043; *see also* Final Office Action, Aug. 1, 2012, at 15; Ex. 1044.)

system divides each of the data streams "into one or more binary objects," having a size of one megabyte or less. (*Id*. at col. 4, ll. 21-30.) The files, and their component binary objects, are then distributed across the network. (*Id*. at col. 3, ll. 24-44.)

To identify and access binary objects, Woodhill creates a Binary Object Identification Record, including a Binary Object Identifier, for each binary object. (*Id*. at col. 7, ll. 60-62.) The Binary Object Identifiers are "calculated from the contents of the data" in the corresponding binary object, and include a hash of the contents of the binary object. (Woodhill at col. 8, ll. 21-24; *see also* col. 7, l. 64 – col. 8, l. 31; Ex. 1005.) As Woodhill emphasizes, the "***critical feature*** to be recognized in creating a Binary Object Identifier 74 is that the ***identifier should be based on the contents of the binary object*** so that the Binary Object Identifier 74 changes when the contents of the binary object changes." (*Id*. at col. 8, ll. 58-62 (emphasis added).) In this way "duplicate binary objects, even if resident on different types of computers in a heterogeneous network, can be recognized from their identical Binary Object Identifiers 74." (*Id*. at col. 8, ll. 62-65.)

Woodhill, like Browne, Langer, and Kantor, specifically addresses the issue of "compound data items." For larger files, the binary objects can be further segmented into "granules" having a size, for example, of one kilobyte. (*Id*. at col. 14, l. 52- col. 15, l. 4.) The "granules" can be used to track changes to binary

objects at the "'granule' level." (*Id*. at col. 14, ll. 62-65; Ex. 1005.) Woodhill calculates a "contents identifier" for each of the granules based on a CRC value and hash value "calculated against the contents of the 'granule.'" (*Id*. at col. 15, ll. 21-28.) He then uses the contents identifiers "to determine what data within a binary object has changed and only back up the changed data instead of the entire binary object." (Woodhill at col. 15, ll. 39-41; Ex. 1005.) Woodhill notes that this "granularization" technique can be "useful when a current version of a file (comprised of current versions of binary objects) must be restored to a previous version of that file (comprised of previous versions of binary objects)." (*Id*. at col. 17, ll. 19-22.) If a binary object is segmented into granules, "[e]ach binary object comprising the current version of the file can be restored to the binary object comprising the previous version of the file by restoring and updating only those 'granules' of the current version of the binary objects that are different between the current and previous versions of the binary objects." (*Id*. at col. 17, ll. 22-27.)

For example, a user may want to restore a large database file to its prior state (or version) of a day before, because of some system malfunction. The system accordingly can save substantial time and bandwidth by limiting the restoration to those granules that have changed. To accomplish this, Woodhill "calculates 'contents identifiers' for each 'granule' within the current version of each binary object as it exists on the local computer 20." (*Id*. at col. 17, ll. 38-40.) He then

requests the previous version of the binary object by "transmit[ing] an 'update request' . . . which includes the Binary Object Identification Record 58 for the previous version of each binary object as well as the list of 'contents identifiers.'" (*Id*. at col. 17, ll. 42-46.) In response to this request, the remote backup file server identifies granules that must be transmitted back to the local computer by reconstituting the previous version of each requested binary object, and comparing the contents identifier for each granule of the reconstituted binary object with the corresponding contents identifiers for the granules transmitted as part of the update request. (Woodhill at col. 17, ll. 46-64; Ex. 1005; *see also* Clark Decl., ¶ 49; Ex. 1009.)

As set forth in detail in the claim chart (Ex. 1048), and as confirmed by Dr. Clark (Clark Decl., ¶¶ 44-59; Ex. 1009), Woodhill in view of Fischer renders obvious each of claims 10 and 21 of the '539 patent. For example, for claims 10 and 21, regarding limitation (A), Dr. Clark confirms that Woodhill discloses a method that, in response to a "request" (an "update request" to restore a current version of a binary object to a prior version) including a "first identifier" (the Binary Object Identifier for the prior version of the binary object), obtains a plurality of "segment identifiers" (the contents identifiers for the granules that have changed relative to the current version of the binary object). (Clark Decl., ¶¶ 48, 50-52; Ex. 1009; Woodhill at col. 7, l. 60 – col. 8, l. 4, col. 17, ll. 17-50; Ex.

1005.) Dr. Clark further confirms that each "segment identifier" (the contents identifiers for the granules) is "based, at least in part, on a function of the data" comprising the granule, and that two identical granules will have identical contents identifiers. (Clark Decl., ¶ 46; Ex. 1009; Woodhill at col. 7, l. 64 – col. 8, l. 1; Ex. 1005.).

Regarding limitations (B) and (C) in claim 10 and limitations (B), (b1), and (b2) in claim 21, Dr. Clark confirms that Woodhill discloses a system composed of "hardware in combination with software" (a system of local and remote computers running the Distributed Storage Manager program), and that, in response to the update request, the "segment identifiers" (the contents identifiers for the granules) are used to request and obtain a particular segment (a granule that is different and should be sent in reply to the request) from one of the computers in the network (the backup server). (Clark Decl., ¶¶ 46, 49, 53; Ex. 1009; Woodhill at col. 17, l. 50 – col. 18, l. 9; Ex. 1005.) Dr. Clark confirms that Woodhill renders obvious "using a particular segment identifier . . . particular segment" in (b0) of claim 21. (Clark Decl., ¶ 54; Ex. 1009; Woodhill at col. 17, l. 50 – col. 18, l. 9; Ex. 1005.)

To the extent PersonalWeb contends that the "first identifier" (the Binary Object Identifier) disclosed in Woodhill is not "based, at least in part, on a second given function of the plurality of segment identifiers," Dr. Clark confirms that this is a mere design choice, and that a person of ordinary skill in the art exercising

ordinary creativity would have found it obvious to modify Woodhill to calculate the "first identifier" based on a function of the granule identifiers, or to combine it with another reference teaching such an identifier. For example, it would have been obvious to create a Binary Object Identifier as a hash of the granule identifiers (*i.e.*, a "hash of hashes).[25] (Clark Decl., ¶ 56; Ex. 1009.) This is a well-known technique, dating back at least to Merkle's work, and has been used in similar contexts by multiple prior art references, including Browne, Langer, Kantor, and Fischer among others. (Clark Decl., ¶ 57; Ex. 1009.) Fischer, for example, builds an aggregate "fileHash" value for a database file containing multiple records by iterating over each record of the file, hashing the data of each record, and hashing the hashes of the records to create the aggregate "fileHash"

[25] For example, a person of ordinary skill exercising ordinary creativity would have found it obvious to combine Woodhill with Fischer to meet that limitation. (Clark Decl., ¶ 56; Ex. 1009.) As Dr. Clark explains, a person of ordinary skill in the art exercising ordinary creativity would have been motivated to modify the Binary Object Identifier calculation to use a technique like the one disclosed by Fischer (or any of the other references disclosing a "hash of hashes," such as Merkle). (Clark Decl., ¶ 59; Ex. 1009.) Calculating Binary Object Identifiers for database files in this way would constitute a simple substitution of one known element for another to obtain predictable results, and therefore it would be obvious to a person of ordinary skill in the art. (Clark Decl., ¶ 59; Ex. 1009.)

(i.e., a "hash of hashes"). (Fischer at col. 7, line 66 – col. 8, line 31; *see also* Abstract; Ex. 1036.) As Dr. Clark explains, a person of ordinary skill in the art, exercising ordinary creativity, would have been motivated to calculate identifiers based on a function of other segment identifiers because this could improve performance. Once contents identifiers are formed, it is faster to compute a "hash of hashes" than to form an identifier by hashing the entirety of the larger object. In addition, if only a relatively small amount of granules change, the operation should result in further performance benefits. (Clark Decl., ¶ 59; Ex. 1009.)

Ground 9: Challenged Claim 34 is Unpatentable as Obvious in view of Woodhill in combination with Browne

Claim 34 recites "dividing" a particular data item into a plurality of segments, as well as determining "segment identifiers" being a "True Name" for each segment, and a data identifier for the data item as a while being a "True Name" of the segment identifiers. Dr. Clark confirms that Woodhill in combination with Browne discloses the limitations "(A) dividing . . . segments" (Woodhill at col. 14, ll. 53-65; Ex. 1005), "(B) determining . . . segment", (Woodhill at col. 15, ll. 20-30; Ex. 1005; Browne at 4, 6; Ex. 1002), "(C) forming . . . identifiers" (Woodhill at col. 5, ll. 20-30; Ex. 1005), "(D) determining . . . item, and" (Woodhill at col. 7, l. 60 – col. 8, l. 4; Ex. 1005; Browne at 4, 6; Ex. 1002), "(E) maintaining . . . item" (Woodhill at col. 7, l. 60 – col. 8, l. 4; Ex. 1005) and

"(F) in response. . . item." (*Id*. at col. 18, ll. 10-19.) (*See* Clark Decl.,¶ 60; Ex. 1009.)

To the extent PersonalWeb contends that "a True Name of the data" is limited to use of a cryptographic hash function such as MD5, a person of ordinary skill in the art would have found it obvious to modify Woodhill to utilize an MD5 hash. Modifying Woodhill's Binary Object Identifiers to use MD5 instead of its disclosed hash function would have been desirable to further reduce the probability of identifier collisions. (Clark Decl., ¶ 60; Ex. 1009.) The use of such an MD5 hash for the same purposes is disclosed in Browne. (*See* Browne at 4, 6; Ex. 1002 (using MD5 as part of the LIFN signatures.) The modification of the Binary Object Identifier calculation to use MD5 would constitute the application of a known technique to a known device, ready for improvement, to yield predictable results, and therefore it would be obvious to a person of ordinary skill in the art. (Clark Decl., ¶ 60; Ex. 1009.) Further, design incentives and market forces would have prompted the application of any of the LIFN Prior Art to Woodhill, because the resulting system would have provided for collision-resistant generation of identifiers using a known function. (Clark Decl., ¶ 60; Ex. 1009.) Finally, the combination would have required no more than ordinary creativity, taking into account the inferences and creative steps that a person of ordinary skill in the art would employ. (Clark Decl., ¶ 60; Ex. 1009.)

VII. CONCLUSION

Based on the foregoing, it is clear that claims 10, 21, and 34 of the '539 Patent recite subject matter that is either anticipated or obvious. The Petitioner requests institution of an *inter partes* review to cancel those claims.

Respectfully Submitted,

/David L. Cavanaugh/

David L. Cavanaugh, Reg. No. 36, 476

1875 Penn. Avenue, NW

Washington DC 2006

CERTIFICATE OF SERVICE

I hereby certify that, on December 16, 2012, I caused a true and correct copy of the foregoing materials:

- Petition for *Inter Partes Review* of U.S. Patent No. 7,945,539
- Exhibits 1001-1048
- Fee Summary Page
- EMC Corp. Power of Attorney

to be served via Federal Express on the following attorney of record as listed on PAIR:

Davidson Berquist Jackson & Gowdey, LLP

Attn: Brian Siritzky, Ph.D.

4300 Wilson Blvd., 7th Floor

Arlington, Virginia 22203

/David L. Cavanaugh/

David L. Cavanaugh

Registration No. 36,476

Table of Exhibits for U. S. Patent 7,945,539 Petition for *Inter Partes* Review

Exhibit	Description
1001.	U.S. Patent No. 7,945,539
1002.	S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," University of Tennessee Technical Report CS-95-278 (Feb. 1995)
1003.	Albert Langer, "Re: dl/describe (File descriptions)," post to the "alt.sources" newsgroup on August 7, 1991
1004.	Kantor, "The Frederick W. Kantor Contents-Signature System Version 1.22," FWKCS122.REF (August 10, 1993)
1005.	Woodhill et al., U.S. Patent No. 5,649,196, entitled "System and Method For Distributed Storage Management on Networked Computer Systems Using Binary Object Identifiers"
1006.	S. Browne et al., "Location-Independent Naming for Virtual Distributed Software Repositories," http://www.netlib.org/utk/papers/lifn/main.html (Nov. 11, 1994)
1007.	K. Moore et al., "An Architecture for Bulk File Distribution," Network Working Group Internet Draft (July 27, 1994)
1008.	Chart of Patent Family Members
1009.	Declaration of Dr. Douglas Clark a Professor of Computer Science at Princeton University
1010.	Banisar et al., The Third CPSR Cryptography and Privacy Conference at 509 (1993)
1011.	G. D. Knott, Hashing functions, The Computer Journal 18 (1975), no. 3, p. 265.
1012.	R. Rivest, "The MD5 Message-Digest Algorithm," Internet RFC 1321 (Apr. 1992)

1013.	McGraw-Hill Dictionary of Scientific and Technical Terms, (4th ed., 1989)
1014.	B. Kaliski, "A Survey of Encryption Standards, " IEEE Micro (Dec. 1993)
1015.	Rabin, Fingerprinting by Random Polynomials, Center for Research in Computing Technology, Harvard University, Report TR-15-81
1016.	U. Manber, "Finding Similar Files in a Large File System", University of Arizonia Technical Report (1994)
1017.	D.R. McGregor and J.A. Mariani 'Fingerprinting' – A Technique for File Identification and Maintenance, Software Practice & Experience 1165 (1982)
1018.	T. Berners-Lee et al., "Uniform Resource Locators (URL)," Internet RFC 1738 (Dec. 1994)
1019.	U. S. Patent 6,415, 280 Prosecution History, Response (August 22, 2001)
1020.	EP Pub. No. EP0826181A1 Prosecution History, Annex to the communication dated May 8, 2009
1021.	EP Pub. No. EP0826181A1 Prosecution History, Reply to communication from the Examining Division dated November 18, 2009
1022.	EP Pub. No. EP0826181A1 Prosecution History, Annex to the communication dated March 14, 2012
1023.	EP Pub. No. EP0826181A1 Prosecution History, Closing of Application dated June 14, 2012
1024.	U.S. Patent 7,945,539 Prosecution History, Application as filed on October 31, 2007
1025.	U.S. Patent 7,945,539 Prosecution History, Office Action of May 6, 2009

1026.	U.S. Patent 7,945,539 Prosecution History, Amendment of October 5, 2009
1027.	U.S. Patent 7,945,539 Prosecution History, Final Rejection of January 12, 2010
1028.	U.S. Patent 7,945,539 Prosecution History, Amendment after Final of March 12, 2010
1029.	U.S. Patent 7,945,539 Prosecution History, Advisory Action, Mar. 23, 2010
1030.	U.S. Patent 7,945,539 Prosecution History, Request for Continued Examination of October 14, 2010
1031.	Merkle, U.S. Patent No 4,309,569, entitled "Method of Providing Digital Signatures," filed Sept. 5, 1979
1032.	Lampson and Sproull, "An Open Operating System for a Single-User Machine," ACM Operating Review (December 1979)
1033.	A. Tanenbaum, "Operating Systems: Design and Implementation", Prentice Hall, (1987)
1034.	E. Babb, "Implementing a Relational Database by Means of Specialized Hardware," ACM Transactions on Database Systems, Vol. 4, No.1, at 2-4, March 1979
1035.	D. Bitton and D. DeWitt, "Duplicate Record Elimination in Large Data Files," ACM Transactions on Database Systems, Vol. 8, No. 2, at 255 – 265 (June 1983)
1036.	A. Fischer, U.S. Patent No. 5,475,826, entitled "Method for Protecting a Volatile File Using a Single Hash," filed Nov. 19, 1993
1037.	R. Williams, "An algorithm for matching text (possibly original)," posted to the "comp.compression" newsgroup on January 27, 1992;
1038.	R. Williams, "An Introduction to Digest Algorithms," Rocksoft

	(Nov. 1994)
1039.	EARN Staff, "Guide to Network Resource Tools," Internet RFC 1580 (March 1994)
1040.	F. Clark, "PCBoard v15.0 Technical Reference Manual," (1993)
1041.	P. Deutsch et al., "How to Use Anonymous FTP," Internet RFC 1635 (May 1994)
1042.	Patent Application 13/352,169, Non-final Office Action dated March 27, 2012
1043.	Patent Application 13/352,169, Response to Non-Final Office Action dated June 26, 2012
1044.	Patent Application 13/352,169, Final Office action dated August 1, 2012
1045.	Claim Chart 1, Invalidity Claim Chart in view of LIFN ("Browne")
1046.	Claim Chart 2, Invalidity Claim Chart in view of Langer
1047.	Claim Chart 3, Invalidity Claim Chart in view of FWKCS Contents – signature System Version 1.22 ("Kantor")
1048.	Claim Chart 4, Invalidity Claim Chart in view of Woodhill